

大学の定期試験時間割編成問題

—南山大学 名古屋キャンパスを例として—

2013SE200 鈴木麻子 2013SE240 山田大輝

指導教員：佐々木美裕

1 はじめに

多くの大学では1年を複数の学期に分け、所定の回数の講義が終わると試験が実施される。通常授業の時間割どおりに定期試験を実施する大学もあるが、通常の授業時間割とは別に定期試験の時間割を試験の実施ごとに作成している大学も多く、定期試験時間割作成者の大きな負担となっている。

大学の定期試験時間割作成の際に考慮すべき最も重要な点は、定期試験期間内にすべての試験を割り当てること、学生および試験監督者が同じ曜日時間に重複しないようにすること、使用する教室は受験者数を収容できる規模のものとするなどである。これらの制約に加え、南山大学ではなるべく講義の曜日時間と同じであることが望ましいと考え、定期試験時間割作成の際にはこのことも考慮する。

現状では、定期試験時間割作成担当者がこれらの制約を考慮しながら手作業で時間割を作成しており、多くの時間を要している。本研究では、定期試験時間割作成担当者の作業負担を減らすことを目標に、オペレーションズリサーチを用いて手作業で行う手間を簡略化することを試みる。

2 過去の研究

小野内 [1] は、南山大学瀬戸キャンパスを例として定期試験時間割編成モデルを提案した。このモデルは定期試験時間割編成問題を2段階に分け、第1段階で試験の時間割を決定し、第2段階で試験室の割り当てと試験監督の割り当てを決定している。この研究では、IBM ILOG CPLEX 12.5を用いて最適化を行っている。

3 瀬戸キャンパスと名古屋キャンパスの違い

南山大学の瀬戸キャンパスと名古屋キャンパスでは、定期試験に関連する規則に違いがあり、大きく3つの違いが挙げられた。

● 試験時間

- 1つ目は名古屋キャンパスの試験時間が原則50分であることである。瀬戸キャンパスは試験時間が80分であるため、同じ時間に50分試験が実施されたとしても50分が経過した時点で試験を終了するので、80分未満の試験と80分の試験を区別せずにわりあてることができた。しかし、名古屋キャンパスでは50分を基準として試験時間割が組まれているため、80分試験は決められた時間に試験を実施するように制約を追加しなければならない。

● 2時限連続で実施する試験の数

- 2つ目は2時限で実施する試験の数である。試験時間が80分よりも長くかかってしまう場合、2時限連続して試験を実施する。名古屋キャンパスでは1限、3限、4限に2時限連続試験を開始することができる。名古屋キャンパスと瀬戸キャンパスの両キャンパスにおいて、90分以上の試験（90分試験、120分試験、150分試験、180分試験）は実施しているが、名古屋キャンパスで実施されている90分以上の試験の数は瀬戸キャンパスで実施されている90分以上の試験の数の約11倍である。瀬戸キャンパスでは2時限連続の試験は数が少なかったためあらかじめ決められた時間に割り当てをしていた。そのため小野内のモデルでは、2時限連続試験のことは考慮されていない。試験を実施する科目は毎年だいたい同じなので、2時限試験のみあらかじめ決めることも1つの方法であるが、あらかじめ試験を実施する時限を決めると、手作業によって決めた割り当てにより、制約式に投入する値が変わり、全体の割り当ても変わってしまう。手作業で2つのパターンを作成した場合、その2つのパターンではデータとして与える値が異なるので、実験後の最適値も異なる。最適値を求めたい場合、手作業で決定した割り当てが最適なものではない。よって、自動で割り当てることにより最適値を導き出すことができる。試験室配置の際には、2時限連続試験を実施する試験室は2時限同じであることを考慮して定式化を行う必要がある。

● 教室の大きさ・配置

- 3つ目は教室の大きさと配置である。瀬戸キャンパスでは同じ棟の同じフロアにはほぼ同じ規模の教室が配置されているが、名古屋キャンパスは棟によってもさらに同じ棟の中でも教室の規模が異なるため、収容人数が大きく異なる。複数の教室で試験を実施する場合、試験監督が教室を巡回することを考慮すると複数の教室間の距離が短いことが望ましい。そこで隣り合う教室の収容人数がバラバラである名古屋キャンパスを扱う本研究では教室のグループ化を行い、複数の教室を使用する試験の試験室の割り当ては1つずつではなく、グループで割り当てを行う。その際、単体の教室であってもその教室を1つのグループとみなすこ

とに注意する.

4 本研究

本研究では小野内 [1] のモデルをベースとし, 南山大学名古屋キャンパスで利用できるようにモデルを変更し, 厳密解を求める Gurobi と短時間で良好な解を探索する SCOP(Solver for CONstraint Programing) を用いた 2 通りの方法で計算実験を行い, 計算時間の違いなどを比較する. 本研究では名古屋キャンパスの教室数がとても多いため, 問題の規模が大きく部屋割り当て問題となり求解が困難になってしまう. そこで名古屋キャンパスの試験を実施することができる教室をグループ化し, 受験者数とグループ化された教室の総収容人数がだいたい同じになるように割り当てていく.

名古屋キャンパスの試験を実施することができる教室を図 1 に示す. 四角のマス 1 つ 1 つは教室を表し, 収容人数はそれぞれ異なる. マスが繋がっている教室同士は同じ棟に配置されていることを表す. 1 つ 1 つの収容人数がだいたい等しく, 近くに配置されている教室同士をグループ化する. 本研究では, 1 つの試験を 1 つの教室グループに割り当てる.

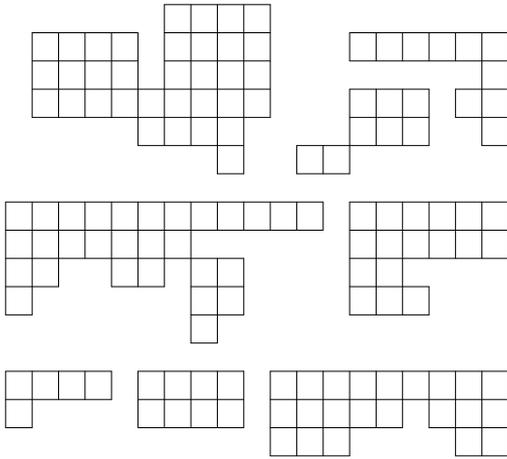


図 1 試験を実施することができる教室

5 定式化

以下に記号を定義する.

T : 試験を実施する曜日時限の添字集合

E : 試験を実施する科目の添字集合

U : 教員の添字集合

T_D : 2 時限連続試験を開始することができる時限の集合

E_D : 2 時限連続試験の集合

E_Z : 50 分または 80 分で実施する試験の集合

m_e : 科目 $e \in E$ の受験者数

f_l : 教室 $l \in L$ の収容人数

c_{te} : 科目 $e \in E$ の試験が曜日時限 $t \in T$ に割り当てられたときのペナルティ

$$p_{tu} = \begin{cases} 1: \text{教員 } u \in U \text{ は曜日時限 } t \in T \text{ に試験監督を} \\ \text{行うことが可能である} \\ 0: \text{上記以外} \end{cases}$$

$$r_{eu} = \begin{cases} 1: \text{教員 } u \in U \text{ は科目 } e \in E \text{ の科目担当教員である} \\ 0: \text{上記以外} \end{cases}$$

$$b_{ij} = \begin{cases} 1: \text{科目 } i \in E \text{ と科目 } j \in E \text{ は同時に試験を} \\ \text{実施することができる} \\ 0: \text{上記以外} \end{cases}$$

$$g_e = \begin{cases} 1: \text{科目 } e \in E \text{ は 80 分試験である} \\ 0: \text{上記以外} \end{cases}$$

$$o_t = \begin{cases} 1: \text{試験時限 } t \in T \text{ は 80 分試験を} \\ \text{実施することができる} \\ 0: \text{上記以外} \end{cases}$$

b_{ij} はすべての学生が同じ曜日時限に 2 つ以上の試験が重複しない時間割を作成する上で, とても重要な役割を果たす. 試験を実施する科目 $i \in E$ と $j \in E$ を両方とも履修している学生が 1 人でもいた場合には, 科目 $i \in E$ と $j \in E$ の試験を同時に実施できない. そのような学生が 1 人もいない場合には, 同時に試験を実施できる.

次に, 以下の決定変数を定義する.

$$w_{te} = \begin{cases} 1: \text{科目 } e \in E \text{ の試験を曜日時限 } t \in T \text{ に実施する} \\ 0: \text{上記以外} \end{cases}$$

$$y_{te} = \begin{cases} 1: \text{試験 } e \in E_D \text{ を時限 } t \in T_D \text{ に開始する} \\ 0: \text{上記以外} \end{cases}$$

y_{te} は 2 時限連続科目を開始する時限のことであり, w_{te} は試験を実施する時限のことである. 2 時限連続科目 $e \in E$ を時限 1 と 2 に実施する場合, $y_{1e} = 1, y_{2e} = 0, w_{1e} = w_{2e} = 1$ となる.

6 第 1 段階

第 1 段階では, 学生が同じ曜日時限に 2 つ以上の試験が重複しないように試験時間の割り当てを行う. 割り当てを行う際には 2 時限連続で実施する試験と 80 分試験を決められた時限に割り当てることを考慮する. また, 小野内 [1] のモデルと異なり, 提案するモデルの第 1 段階では, 計算時間短縮のために教室割り当てを行わない.

以上の記号を用いて次のように定式化する.

$$\text{Minimize} \quad \sum_{t \in T} \sum_{e \in E} c_{te} w_{te} \quad (1)$$

subject to

$$w_{te} + r_{eu} \leq p_{tu} + 1, \quad t \in T, e \in E, u \in U \quad (2)$$

$$w_{ti} + w_{tj} \leq b_{ij} + 1, \quad t \in T, i \in E, j \in E \quad (3)$$

$$g_e + w_{te} \leq o_t + 1, \quad t \in T, e \in E \quad (4)$$

$$\sum_{e \in E} m_e w_{te} \leq 0.8 \sum_{l \in L} f_l, \quad t \in T \quad (5)$$

$$\sum_{t \in T_D} y_{te} = 1, \quad e \in E_D \quad (6)$$

$$w_{te} + w_{(t+1)e} \geq 2y_{te}, \quad t \in T_D, e \in E_D, e \in E_D \quad (7)$$

$$\sum_{t \in T} w_{te} = 1, \quad e \in E_Z \quad (8)$$

$$\sum_{t \in T_D} w_{te} = 2, \quad e \in E_D \quad (9)$$

$$w_{te} \in \{0, 1\}, \quad t \in T, e \in E \quad (10)$$

$$y_{te} \in \{0, 1\}, \quad t \in T_D, e \in E \quad (11)$$

目的関数 (1) は、ペナルティの総和を表す。式 (2) は、各科目の試験は科目担当教員の都合の良い曜日時限に実施することを表す。式 (3) は、各学生は同時に複数の科目の試験を受験できないことを表す。式 (4) は、80 分試験を 80 分試験が実施できる時限に割り当てて表す。式 (5) は、ある時限に実施される総受験者数はすべての教室の収容人数の総和の 80% 以下であることを表す。式 (6) は、2 時限連続試験は各科目 1 回ずつ実施することを表す。式 (7) は、2 時限連続試験を 2 時限連続で実施することを表す。式 (8) は、1 時限で実施される試験はそれぞれ 1 回ずつ実施することを表す。式 (9) は、2 時限連続科目はそれぞれ 2 時限ずつ試験を実施することを表す。式 (10)(11) は、0-1 変数であることを表す。

7 SCOP での解法

SCOP では、試験室と監督教員のことは考慮せず、試験科目の曜日時限の決定のみを行う。はじめに、試験科目の曜日時限を決定するために以下の記号を定義する。

T : 試験を実施する曜日時限の添字集合

D : 試験日の添字集合

E : 試験を実施する科目の添字集合

c_{te} : 科目 $e \in E$ の試験が曜日時限 $t \in T$ に割り当てられたときのペナルティ

$$b_{ij} = \begin{cases} 1: \text{科目 } i \in E \text{ と科目 } j \in E \text{ は同時に試験を} \\ \text{実施することができる} \\ 0: \text{上記以外} \end{cases}$$

次に、以下の決定変数を定義する。

$$w_{te} = \begin{cases} 1: \text{科目 } e \in E \text{ の試験を曜日時限 } t \in T \text{ に実施する} \\ 0: \text{上記以外} \end{cases}$$

以上の記号を用いて次のように定式化する。

$$\text{Minimize} \quad \sum_{t \in T} \sum_{e \in E} c_{te} w_{te} \quad (12)$$

subject to

$$\sum_{t \in T} w_{te} = 1, \quad e \in E \quad (13)$$

$$w_{ti} w_{tj} \leq b_{ij}, \quad i \in E, j \in E, t \in T \quad (14)$$

目的関数 (12) はペナルティの総和を表す。式 (13) は、各科目は 1 回ずつ試験を実施することを表す。式 (14) は、各学生は同時に複数の試験を受験できないことを表す。

8 第 2 段階

第 2 段階では、第 1 段階で決定した時間割を用いて、受験者数と教室の収容人数がだいたい同じになるように試験室の決定を行う。はじめに、第 2 段階で用いる記号を定義する。

K_s : 教室グループ s , $K_s \subseteq L$, $s \in N$

L : 教室の集合

N : 教室のグループ数

A_s : 教室グループ $s \in N$ の収容人数

$$v_{sl} = \begin{cases} 1: \text{教室グループ } s \in N \text{ に教室 } l \in L \text{ が含まれる} \\ 0: \text{上記以外} \end{cases}$$

次に、以下の決定変数を定義する。

$$x_{tes} = \begin{cases} 1: \text{教室 } e \in E \text{ の試験を曜日時限 } t \in T \text{ に} \\ \text{教室グループ } s \in N \text{ で実施する} \\ 0: \text{上記以外} \end{cases}$$

$$z_{es} = \begin{cases} 1: \text{科目 } e \in E \text{ を教室グループ } s \in N \text{ で実施する} \\ 0: \text{上記以外} \end{cases}$$

以上の記号を用いて、第 2 段階の問題を定式化すると以下の通りとなる。

$$\text{Minimize} \quad \sum_{e \in E} \sum_{t \in T} \sum_{s \in N} A_s x_{tes} - \sum_{t \in T} \sum_{e \in E} m_e w_{te} \quad (15)$$

subject to

$$x_{tes} \leq A_s, \quad t \in T, e \in E, s \in N \quad (16)$$

$$\sum_{s \in N} x_{tes} = w_{te}, \quad t \in T, e \in E \quad (17)$$

$$\sum_{s \in N} \sum_{e \in E} v_{sl} x_{tes} \leq 1, \quad t \in T, l \in L \quad (18)$$

$$\sum_{s \in N} (x_{tes} + x_{(t+1)es}) \geq y_{te}, \quad t \in T_D, l \in L, e \in E_D \quad (19)$$

$$\sum_{s \in N} z_{es} = 1, \quad e \in E \quad (20)$$

$$\sum_{t \in T} x_{tes} \leq 2z_{es}, \quad e \in E_D, s \in N \quad (21)$$

$$x_{tes} \in \{0, 1\}, \quad t \in T, e \in E, s \in N \quad (22)$$

$$z_{es} \in \{0, 1\}, \quad s \in N, e \in E \quad (23)$$

目的関数 (15) は割り当てられた試験室の空席の総和を表す。式 (16) は受験者数が試験室の収容人数以下であることを表す。式 (17) は全ての試験をいずれかの教室グループまたは試験室に割り当てる。式 (18) は試験室が重複しないことを表す。式 (19) は 2 時限連続試験を 2 時限同じ試験室で試験を実施することを表す。式 (20) は 1 つの教室グループで 1 つの試験を実施することを表す。式 (21) は 2 時限連続試験のみ 2 時限分試験を実施することを表す。式 (22)(23) は 0-1 変数であることを表す。

9 計算結果と考察

使用した最適化ソフトウェアは Gurobi6.5.2 であり、計算環境は (プロセッサ: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.40GHz 実装メモリ: 16GB) である。計算には、2016 年度南山大学名古屋キャンパス春学期の約半分である、月曜日 1 限から 6 限、火曜日 1 限から 6 限、水曜日 1 限、2 限の実データを使用して第 1 段階の問題の計算を行った。来年度から南山大学にクォーター制度が導入されるため、1 クォーターで試験を実施する数はセメスターの約半分と考えられるためである。

第 1 段階の問題を解くにあたりペナルティを次のように設定した。講義時間割と同じ曜日時限に実施する場合のペナルティは 0 とする。講義時限が 1 限の場合、同じ曜日の 2,3 限に実施するときのペナルティを 5 とし、4,5,6 限に実施するときのペナルティを 10 とする。講義時限が 1 限以外の場合は、同じ曜日で実施する時限と講義時限との差が 1 のときのペナルティを 5 とし、差が 2 以上であるときのペナルティを 10 とする。同じ曜日のその他時限に実施する場合のペナルティだけ 100 とする。そして、曜日が異なる場合のペナルティは全て 120 とした。試験を実施する曜日時限数 42, 科目数 461, 科目数のうち 2 時限連続科目数 30, 教室のグループ数 573, 変数の数は 20,622, 計算時間は 4,635 秒, 最適値は 14,840 であった。2 時限連続試験は、2 時限のうちどちらかの時限が講義時限とかぶっている場合、同じであると判断する。第 1 段階では、全体の 98.6% が、講義時間割と同じ曜日時限に試験を割り当てられている。この結果より、ほとんどの科目が講義時間割と同じ曜日時限に割りあてることができた。

第 2 段階の結果を考察する。第 2 段階では空席の数が 2416, 計算時間は 31,56 秒となった。試験を実施する教室は普段授業が行われている教室とは異なり、棟も異なるためあまり良い結果であるとはいえない。各計算時間は、第 1 段階が 5 分弱、第 2 段階は 30 秒強であるため、十分に実用可能なレベルであると考ええる。

表 1 講義時間割と同じ曜日時限に試験を実施する科目の割合

2 時限連続試験	100%
2 時限連続試験以外	98.5%
全ての科目	98.6%

SCOP の計算結果を考察する。試験を実施する曜日時限数 42, 科目数 717, 2 時限連続試験と 80 分試験は 50 分試験と区別しないものとする。ペナルティは第 1 段階の決定の際と同じ設定方法であり、計算時間は 10 分、最適値は 16,860 であった。試験科目のうち、その他時限と土曜日に割り当てられた試験は全体の約 10% であった。この結果より、普段講義が行われていない曜日や時限に割り当てられている試験科目が少ないことがわかる。同じ制約式を用いて Gurobi で計算実験を行ったところ、計算時間が 27.1 分かかったが、最適値は出力されなかった。制約式が少ないため、最適値の候補が多いからだと考える。Gurobi と SCOP を比較すると、SCOP は計算時間も短く、試験時間割を作成することができた。

参考文献

- [1] 小野内雄一:『大学の定期試験時間割編成モデルの研究』, 南山大学大学院数理情報研究科 修士論文, 2014.
- [2] 久保幹夫, ジョア ペドロ ペドロソ, 村松正和, アブドゥール レイス:『あたらしい数理最適化 Python 言語と Gurobi で解く』. 近代科学社, 東京, 2012.
- [3] 『Logistics Optimization』,
<<http://logopt.com/scop.htm>>(参照 2016-11-25)